

Base de Datos II - Tarea de Laboratorio No.06. -

SQL DML en Oracle Database

Estudiante: Iris Marisol Hanampa Bellido

Enunciado

Se desea tener una base de datos que almacene la información sobre los empleados de una empresa, los departamentos en los que trabajan y los estudios de que disponen. Guardaremos el historial laboral y salarial de todos los empleados.

Realizar las siguientes operaciones:

1. Los siguientes atributos son obligatorios:

- NOMBRE (en todas las tablas),
- APELLIDO1 en EMPLEADOS,
- PRESUPUESTO en DEPARTAMENTOS,
- SALARIO en HISTORIAL_SALARIAL
- SALARIO_MIN y SALARIO_MAX en TRABAJOS.

Al crear las tablas correspondientes especificar la opción NOT NULL.

```
CREATE TABLE Departamentos (
    dpto_cod NUMBER(5) PRIMARY KEY,
    nombre_dpto VARCHAR2(30) NOT NULL,
    dpto_padre NUMBER(5),
    presupuesto NUMBER NOT NULL,
    pres_actual NUMBER,
    CONSTRAINT fk_dpto_padre FOREIGN KEY (dpto_padre)
        REFERENCES Departamentos(dpto_cod)
);
```

```
CREATE TABLE Empleados (
    dni NUMBER(8) PRIMARY KEY,
    nombre VARCHAR2(10) NOT NULL,
    apellido1 VARCHAR2(15) NOT NULL,
    apellido2 VARCHAR2(15),
    direcc1 VARCHAR2(25),
    direcc2 VARCHAR2(20),
    ciudad VARCHAR2(20),
    provincia VARCHAR2(20),
    cod_postal VARCHAR2(5),
    sexo VARCHAR2(1),
```

```
fecha_nac DATE,  
dpto_cod NUMBER(5),  
CONSTRAINT fk_empleado_departamento FOREIGN KEY (dpto_cod)  
    REFERENCES Departamentos(dpto_cod)  
);
```

```
CREATE TABLE Universidades (  
    univ_cod NUMBER(5) PRIMARY KEY,  
    nombre_univ VARCHAR2(25) NOT NULL,  
    ciudad VARCHAR2(20),  
    municipio VARCHAR2(2),  
    cod_postal VARCHAR2(5)  
);
```

```
CREATE TABLE Estudios (  
    empleado_dni NUMBER(8),  
    universidad NUMBER(5),  
    año NUMBER,  
    grado VARCHAR2(3),  
    especialidad VARCHAR2(20),  
    CONSTRAINT fk_estudios_empleado FOREIGN KEY (empleado_dni)  
        REFERENCES Empleados(dni),  
    CONSTRAINT fk_estudios_universidad FOREIGN KEY (universidad)  
        REFERENCES Universidades(univ_cod)  
);
```

```
CREATE TABLE Trabajos (  
    trabajo_cod NUMBER(5) PRIMARY KEY,  
    nombre_trab VARCHAR2(20) NOT NULL,  
    salario_min NUMBER(2) NOT NULL,  
    salario_max NUMBER(2) NOT NULL  
);
```

```
CREATE TABLE Historial_Laboral (  
    empleado_dni NUMBER(8),  
    univ_cod NUMBER(5),  
    trabajo_cod NUMBER(5),  
    fecha_inicio DATE,  
    fecha_fin DATE,  
    dpto_cod NUMBER(5),  
    supervisor_dni NUMBER(8),  
    CONSTRAINT pk_historial_laboral PRIMARY KEY (empleado_dni, trabajo_cod,  
    fecha_inicio),  
    CONSTRAINT fk_historial_laboral_empleado FOREIGN KEY (empleado_dni)  
        REFERENCES Empleados(dni),  
    CONSTRAINT fk_historial_laboral_universidad FOREIGN KEY (univ_cod)  
        REFERENCES Universidades(univ_cod),  
    CONSTRAINT fk_historial_laboral_trabajo FOREIGN KEY (trabajo_cod)
```

```

    REFERENCES Trabajos(trabajo_cod),
CONSTRAINT fk_historial_laboral_dpto FOREIGN KEY (dpto_cod)
    REFERENCES Departamentos(dpto_cod),
CONSTRAINT fk_historial_laboral_supervisor FOREIGN KEY (supervisor_dni)
    REFERENCES Empleados(dni)
);

CREATE TABLE Historial_Salarial (
    empleado_dni NUMBER(8),
    salario      NUMBER NOT NULL,
    fecha_comienzo DATE,
    fecha_fin    DATE,
    CONSTRAINT pk_historial_salarial PRIMARY KEY (empleado_dni,
fecha_comienzo),
    CONSTRAINT fk_historial_salarial_empleado FOREIGN KEY (empleado_dni)
    REFERENCES Empleados(dni)
);

```

2. **El atributo SEXO en EMPLEADOS sólo puede tomar los valores H para hombre y M para mujer.**

```

ALTER TABLE Empleados
ADD CONSTRAINT check_empleado_sexo CHECK(sexo IN ('H', 'M') );

```

3. **Dos DEPARTAMENTOS no se llaman igual. Dos TRABAJOS tampoco.**

```

ALTER TABLE Departamentos
ADD CONSTRAINT unique_departamento_nombre UNIQUE(nombre_dpto);

```

```

ALTER TABLE Trabajos
ADD CONSTRAINT unique_trabajo_nombre UNIQUE(nombre_trab);

```

4. **Cada empleado tiene un solo salario en cada momento. También, cada empleado tendrá asignado un solo trabajo en cada momento.**

```

ALTER TABLE Historial_Salarial
ADD CONSTRAINT unique_salario_por_periodo UNIQUE (empleado_dni,
fecha_comienzo);

```

```

ALTER TABLE Historial_Laboral
ADD CONSTRAINT unique_trabajo_por_periodo
UNIQUE (empleado_dni, fecha_inicio);

```

5. **Se ha de mantener la regla de integridad de referencia y pensar una clave primaria para cada tabla.**

Ya se encuentra definido en el script de creación

6. **Agregue a la tabla empleados los campos de teléfono y celular para tener como ubicar rápidamente al empleado.**

```
ALTER TABLE Empleados
ADD (
    telefono VARCHAR2(15),
    celular VARCHAR2(15)
);
```

- 7. Inserte las siguientes filas (las columnas que no aparecen se considerarán nulas).**

```
INSERT INTO Empleados (nombre, apellido1, apellido2, dni, sexo)
VALUES ('Sergio', 'Palma', 'Entrena', 111222, 'H');
```

```
INSERT INTO Empleados (nombre, apellido1, apellido2, dni, sexo)
VALUES ('Ana', 'Torres', 'Mendoza', 111223, 'M');
```

```
INSERT INTO Historial_Laboral (empleado_dni, fecha_inicio, dpto_cod)
VALUES (111222, TO_DATE('16/06/1996','DD/MM/YYYY'), 222333);
```

- 8. ¿Qué ocurre si se modifica esta última fila de historial_laboral asignándole al empleado 111222 un supervisor que no existe en la tabla de empleados?**

Saltaría un error de integridad referencial al momento de ejecutar el update indicando que la clave ingresada debe existir para continuar con la operación.

- 9. Borre una universidad de la tabla de UNIVERSIDADES ¿Qué le sucede a la restricción de clave ajena de la tabla ESTUDIOS? Altere la definición de la tabla para que se mantenga la restricción, aunque se borre una universidad.**

Actualmente como está definido, no permitirá el borrado de registros dado que se encuentran asociados con otros por la clave foránea, no obstante se puede modificar para que sí lo permita.

a) Borra también los estudios relacionados con esa universidad.

```
ALTER TABLE ESTUDIOS
DROP CONSTRAINT fk_estudios_universidad;
ALTER TABLE ESTUDIOS
ADD CONSTRAINT fk_estudios_universidad
FOREIGN KEY (id_universidad)
REFERENCES UNIVERSIDADES(id_universidad)
ON DELETE CASCADE;
```

b) Si se borra una universidad, el campo id_universidad quedará NULL en Estudios.

```
ALTER TABLE Estudios
DROP CONSTRAINT fk_estudios_universidad;
```

```
ALTER TABLE Estudios
ADD CONSTRAINT fk_estudios_universidad
FOREIGN KEY (id_universidad)
REFERENCES Universidades(id_universidad)
```

ON DELETE SET NULL;

- 10. Añada una restricción que obligue a que las personas que hayan introducido la CIUDAD deban tener el campo COD_POSTAL a NOT NULL. ¿Qué ocurre con las filas ya introducidas?**

```
ALTER TABLE Empleados  
ADD CONSTRAINT chk_ciudad_codpostal  
CHECK(ciudad IS NULL OR cod_postal IS NOT NULL);
```

En base a las filas ya introducidas, se verificará internamente si estas ya cumplen con la restricción, en caso contrario, dará un fallo el alter table y no podrá aplicarse. Por ende, primero se debería dar una corrección en los datos antes de su ejecución.

- 11. Añada un nuevo atributo VALORACIÓN en la tabla de EMPLEADOS que indique de 1 a 10 la valoración que obtuvo el empleado en su entrevista de trabajo al iniciar su andadura en la empresa. Ponga el valor por defecto 5 para ese campo.**

```
ALTER TABLE Empleados  
ADD valoracion NUMBER(2)  
DEFAULT 5  
CONSTRAINT chk_valoracion CHECK (valoracion BETWEEN 1 AND 10);
```

- 12. Elimine la restricción de que el atributo NOMBRE de la tabla EMPLEADOS no puede ser nulo.**

```
ALTER TABLE Empleados  
MODIFY nombre VARCHAR2(10) NULL;
```

- 13. Modificar el tipo de datos de DIREC1 de la tabla EMPLEADOS a cadena de caracteres de 40 como máximo.**

```
ALTER TABLE Empleados  
MODIFY direcc1 VARCHAR2(40);
```

- 14. ¿Podría modificar el tipo de datos del atributo FECHA_NAC de la tabla EMPLEADOS Y convertirla a tipo cadena?**

```
ALTER TABLE Empleados  
MODIFY fecha_nac VARCHAR2(10);
```

- 15. Cambiar la clave primaria de EMPLEADOS al NOMBRE y los dos APELLIDOS.**

```
ALTER TABLE Empleados  
DROP PRIMARY KEY;
```

```
ALTER TABLE Empleados  
ADD CONSTRAINT pk_empleados PRIMARY KEY (nombre, apellido1, apellido2);
```

- 16. Crear una nueva tabla llamada INFORMACIÓN UNIVERSITARIA que tenga el NOMBRE y los dos APELLIDOS (en un solo atributo) de todos los**

EMPLEADOS junto con la UNIVERSIDAD donde estudiaron. Cárgala con los datos correspondientes.

```
CREATE TABLE Informacion_Universitaria (
    empleado_nombre_completo VARCHAR2(100),
    universidad      VARCHAR2(25)
);
```

```
INSERT INTO Informacion_Universitaria (empleado_nombre_completo, universidad)
SELECT RTRIM(e.nombre || ' ' || e.apellido1 || ' ' || NVL(e.apellido2, '')),
       u.nombre_univ
  FROM Empleados e
 JOIN Estudios est
    ON e.dni = est.empleado_dni
 JOIN Universidades u
    ON est.universidad = u.univ_cod;
```

- 17. Crear una vista llamada NOMBRE_EMPLEADOS con el NOMBRE y los dos APELLIDOS (en un solo atributo) de todos los EMPLEADOS que son de Málaga.**

```
CREATE VIEW Nombre_Employados AS
SELECT
    nombre || ' ' || apellido1 || ' ' || NVL(apellido2, '') AS nombre_completo
  FROM Empleados
 WHERE ciudad = 'Málaga';
```

- 18. Crear otra vista llamada INFORMACION_EMPLEADOS con el NOMBRE y los dos APELLIDOS (en un solo atributo) y EDAD (no fecha de nacimiento) de todos los EMPLEADOS.**

```
CREATE VIEW Informacion_Employados AS
SELECT
    nombre || ' ' || apellido1 || ' ' || NVL(apellido2, '') AS nombre_completo,
    TRUNC(MONTHS_BETWEEN(SYSDATE, fecha_nac) / 12) AS edad
  FROM Empleados;
```

- 19. Crear otra vista sobre la anterior llamada INFORMACION_ACTUAL que dispone de toda la información de INFORMACION_EMPLEADOS junto con el SALARIO que está cobrando en este momento.**

```
CREATE VIEW Informacion_Actual AS
SELECT
    ie.nombre_completo,
    ie.edad,
    e.salario
  FROM Informacion_Employados ie
 JOIN Empleados e
    ON ie.nombre_completo = e.nombre || ' ' || e.apellido1 || ' ' || NVL(e.apellido2, '');
```

20. Borrar todas las tablas. ¿Hay que tener en cuenta las claves ajenas a la hora de borrar las tablas?

Sí hay que tener en cuenta las claves ajena, porque determinan el orden de borrado y porque existe una restricción de integridad referencial.

- Borrar primero las tablas dependientes y luego las independientes

```
DROP TABLE Historial_Laboral;  
DROP TABLE Historial_Salarial;  
DROP TABLE Estudios;  
DROP TABLE Empleados;  
DROP TABLE Trabajos;  
DROP TABLE Universidades;  
DROP TABLE Departamentos;
```